IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appeal Brief Transmittal

AFW

In re application of: Miller

Serial No.: 09/780,196

Filed on: 02/09/01

For: **APPARATUS AND METHOD FOR TIMEOUT-FREE WAITING FOR AN ORDERED MESSAGE IN A CLUSTERED COMPUTING ENVIRONMENT**

Mail Stop APPEAL BRIEF - PATENT
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Transmitted herewith for filing is an **Appeal Brief** in triplicate for the above-identified Application.

[X] Please deduct $500.00 from Deposit Account No. 09-0465 for IBM Corporation to cover the fee under 37 C.F.R. §1.17(f) for the filing of the enclosed appeal brief. A duplicate copy of this sheet is enclosed.

[X] The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account No. 09-0465 for IBM Corporation. A duplicate copy of this sheet is enclosed.

[X] Any additional filing fees required under 37 C.F.R. §1.16.

[X] Any patent application processing fees under 37 C.F.R. §1.17.

MARTIN & ASSOCIATES, L.L.C.
P.O. Box 548
Carthage, MO 64836-0548
(417) 358-4700
FAX (417) 358-5757

Respectfully submitted,

By Derek P. Martin

Derek P. Martin
Reg. No. 36,595

I HEREBY CERTIFY THAT THE CORRESPONDENCE TO WHICH THIS STATEMENT IS AFFIXED IS BEING DEPOSITED WITH THE UNITED STATES POSTAL SERVICE, POSTAGE PAID, AS FIRST CLASS MAIL IN AN ENVELOPE ADDRESSED TO: MAIL STOP APPEAL BRIEF - PATENT, COMMISSIONER FOR PATENTS, P.O. BOX 1450, ALEXANDRIA, VA 22313-1450.

Date: August 30, 2005 By: Derek P. Martin

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Miller	Docket No.:	ROC919990110US2
Serial No.:	09/780,196	Group Art Unit:	2157
Filed:	02/09/01	Examiner:	ALAM, UZMA

For: APPARATUS AND METHOD FOR TIMEOUT-FREE WAITING FOR AN ORDERED MESSAGE IN A CLUSTERED COMPUTING ENVIRONMENT

APPEAL BRIEF

Mail Stop APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir/Madam:

This appeal is taken from the Examiner's final rejection, set forth in the Office Action dated 04/05/05, of appellant's claims 1-22. Appellant's Notice of Appeal under 37 C.F.R. § 1.191 was mailed on 07/05/05.

REAL PARTY IN INTEREST

International Business Machines Corporation is the Real Party in Interest.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences for this patent application. One of the two parent patent applications is currently being appealed, namely serial no. 09/421,585 filed on 10/22/99.

STATUS OF CLAIMS

As filed, this case included claims 1-22. In the pending rejection, claims 1-22 were rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,247,059 to Johnson *et al.* (hereinafter "Johnson"). No claim was allowed. No claims have been added, amended, or cancelled. Claims 1-22 as originally filed are currently pending.

STATUS OF AMENDMENTS

In response to the first office action dated 08/24/04, a Request for Reconsideration was filed on 11/23/04. No amendment has been filed. Therefore, the claims at issue in this appeal are claims 1-22 as originally filed.

SUMMARY OF INVENTION

A clustered computer system includes multiple computer systems (or nodes) on a network that can become members of a group to work on a particular task, referred to herein as a protocol. A protocol is defined so that each phase of the protocol is terminated with an acknowledge (ACK) round. Within each phase of the protocol, a node cannot both send and receive a data message. The protocol includes an ACK round that provides a relative time event that indicates when a data message should be received. If the data message is not received when the ACK round occurs, the receiver knows that the sender did not send it, and can request that the sender re-send the missing data message. In a first embodiment, referred to herein as the "post-ACK" case, the receipt of expected data messages is checked after the ACK round occurs. In a second embodiment, referred to herein as the "pre-ACK" case, the receipt of expected data messages is checked before the ACK round occurs. In both cases, when the receiver sees the ACK round, it knows that the sender sent the data message. If the data message was not received by the receiver, it knows to request that the sender re-send the data message. In this manner the preferred embodiments provide an architected way for sending and receiving data messages without using timers in a clustered computing environment that includes ordered messages. As a result, the preferred embodiments may be readily implemented on computer clusters that include nodes on a wide area network (WAN) that have a large variability in their response times.

ISSUES

The following single issue is presented for review on this Appeal:

1. **Whether claims 1-22 are unpatentable under 35 U.S.C. §102(e) as being anticipated by Johnson**

GROUPING OF CLAIMS

Claims 1, 5, 6, 10, 11, 15-18 and 22 are grouped, and stand or fall together based on claim 1. Claims 2, 7, 12 and 19 are grouped, and stand or fall together based on claim 2. Claims 3, 8, 13 and 20 are grouped, and stand or fall together based on claim 3. Claims 4, 9, 14 and 21 are grouped, and stand or fall together based on claim 4. This grouping of claims is appropriate because each set of claims includes a unique combination of limitations not found in the other sets of claims.

ARGUMENT

Issue 1: Whether claims 1-22 are unpatentable under 35 U.S.C. §102(e) as being anticipated by Johnson

Claim 1

In the final Office Action dated 04/05/05, the Examiner rejected claims 1-22 under 35 U.S.C. §102(e) as being anticipated by Johnson. In the rejection, the Examiner states that Johnson teaches the protocol recited in claim 1, citing col. 2 lines 26-43; col. 5, lines 42-67; and col. 6, lines 1-7 of Johnson. The Examiner further states that Johnson teaches the job recited in claim 1, citing col. 2, lines 26-43 and col. 6, lines 1-34 of Johnson. Appellant respectfully asserts that Johnson does not teach or suggest the protocol nor the job specified in claim 1, and furthermore expressly teaches away from these limitations.

Claim 1 includes the following limitation:

> . . . a protocol residing in the memory that specifies at least one data message and at least one acknowledge (ACK) round that provides a time benchmark for determining whether or not a message has been received without using any timer; . . .

The language cited by the Examiner at col. 2 lines 34-40 of Johnson states:

> A failure by the sender node to receive an ACK message from any of the nodes for whom the multicast message was intended within an allotted time period will prompt the sender to assume that the non-responding receiver node(s) did not, for whatever reason, receive the multicast message, and to begin sending point-to-point messages to such nodes.

The phrase "within an allotted time period" in the examiner's own rejection language shows that Johnson uses a timer while awaiting ACK messages. Col. 6 lines 38-41 of

Johnson states: "The sender node 12 . . . will then set a timer and move to step 74 where it will wait for receipt of acknowledgments (ACKs) . . ." This language shows conclusively that Johnson uses a timer while awaiting receipt of acknowledgments (ACKs). The ACK round in claim 1 "provides a time benchmark for determining whether or not a message has been received without using any timer". Because Johnson expressly teaches the use of a timer while awaiting acknowledgments, Johnson cannot read on the protocol recited in claim 1.

In the Response to Arguments section of the final office action, the examiner addresses the appellant's argument above by stating the following:

> Applicant argues that the reference teaches the use of a timer and therefore does not anticipate the limitations of claims [sic] 1. in [sic] response to the argument, Examiner points out that the reference, Johnson et al. US Patent No. 6,247,059 does not use the timer in determining timing benchmarks for the ACK messages. The system does have a timer, but it is not used in the point-to-point messaging between the sender and the receiver and is not used to determine whether a message has been received or not, as stated in the claim. The messages are sent until an ACK message is received from the receiving node. See column 2, lines 39-60 and column 7, lines 24-67, column 8, lines 1-20 for further detail.

This language by the examiner illustrates a misunderstanding on the examiner's part regarding the teachings of Johnson. The examiner states that Johnson does not use the timer in determining timing benchmarks for the ACK messages. This language by the examiner does not correspond to the language in claim 1, which recites:

> . . . a protocol residing in the memory that specifies at least one data message and at least one acknowledge (ACK) round that provides a time benchmark for determining whether or not a message has been received without using any timer; and
>
> a job residing in the memory and executed by the at least one processor, the job processing the protocol, wherein the job functions according to receiver logic that uses the at least one ACK round to

determine without using any timer whether the at least one data message has been received.

Note that the term "without using any timer" occurs in both of the above-quoted clauses in claim 1. We now determine whether Johnson teaches the limitations of these clauses without using any timer.

The first quoted limitation above is a protocol that specifies at least one data message and at least one acknowledge (ACK) round that provides a time benchmark for determining whether or not a message has been received without using any timer. Note that in the rejection, the examiner has not identified what teaching in Johnson corresponds to the protocol in claim 1. The examiner simply cites to portions of Johnson without indicating their relevance to the many limitations recited in the claim. For this reason, the examiner has failed to establish a prima facie case of anticipation for claim 1 under 35 U.S.C. §102(e).

Johnson teaches the sending of a data message in a computer cluster, and the acknowledgment of the message by an ACK message sent by the receiving computer systems. Note, however, that the protocol recited in claim 1 resides in the memory and is processed by the job, and specifies at least one data message and at least one ACK round that provides a time benchmark for determining whether or not a message has been received without using any timer. There is nothing in Johnson that resides in memory and is processed by a job, and that specifies at least one data message and at least one ACK round that provides at time benchmark for determining whether or not a message has been received without using any timer. FIG. 8 shows one suitable example of a protocol within the scope of claim 1, which defines multiple phases that include data messages, with each phase followed by an ACK round. The ACK rounds in the protocol of FIG. 8 provide a time benchmark for determining whether or not a message has been received without using any timer, as expressly recited in claim 1. Appellant forcefully submits that the messages and ACK rounds in Johnson do not read on the protocol recited

in claim 1, and that the timer in Johnson that is used while awaiting ACK messages precludes application of Johnson to these two claim limitations that expressly recite "without using any timer."

Johnson is one specific example of the prior art that is discussed in detail in appellant's specification. Appellant's Background of the Invention section at p. 3, lines 1-25 states:

> Processing tasks in a computer cluster that uses ordered messages requires that each node process the same task (known as a "protocol"). When a point in the protocol is reached where one node requires a data message from another node, the node that expects the data message (the "receiver") typically configures a timer to wait on the expected data message. If the expected data message is received before the timer times out, the data message is processed normally. If the timer times out before the expected data message is received, an error has occurred. In the prior art, great effort has been expended on defining suitable timeout values that will cause the timer to time out when an error occurs but not under normal operating conditions. Tweaking the timeout values may provide acceptable results for a local area network (LAN), where the time between sending and receiving a message varies within known limits. However, when a computer cluster includes nodes that are coupled via a wide area network (WAN), the tuning of the timeout values becomes very problematic. As the load on the individual LANs coupled to the WAN varies, the time between sending and receiving a message can vary greatly. In this environment, the node that is expecting a data message has to decide what action to take when the timer times out. If the timer times out due to abnormally high network traffic, but the expected data message was actually sent, how does the receiver handle the data message that is received after the timer times out? When the timer times out, the receiver has no idea whether the expected data message was sent by the sender or not. One way to handle a timeout is for the receiver to request that the sender re-send the data message. However, if the original data message was sent but arrives after the timer times out, how does the receiver know whether the data message is the original message or the re-sent message? And if it's the original message, how does the receiver handle the re-sent message when it is received? Providing a timeout timer for a receiver that expects a data message thus presents many problems that are not adequately addressed by the prior art.

Johnson is one example of this type of prior art system described above. Referring to FIG. 3 of Johnson, a timer is set during step 70 when a multicast message is sent. See Johnson, col. 6 lines 33-46. Col. 7 lines 9-22 of Johnson states:

> The timer (not shown) that is set in the multicast message send step 70 will eventually time out, if all the intended receiving nodes of a multicast transmission have not acknowledged the transmission within the determined run time computed round-trip interval. Thereafter, the multicast sender node 12 will begin resending the multicast message point-to-point, to those nodes not yet providing ACK or NAK messages (steps 82-84), and continue doing so until the sender node 12 receives an ACK message from all the nodes or the non-respond node(s) 12 is declared to have been removed from the system (step 82). When all nodes have responded (or those that haven't are declared to no longer by in the system) the multicast transmission will considered to be concluded (step 86).

This language shows conclusively that the examiner's interpretation of Johnson is incorrect. In trying to bolster the assertion that Johnson does not use a timer, the examiner states that messages in Johnson are sent until an ACK message is received from the receiving node. The quoted language above from Johnson shows that this assertion by the examiner is incorrect. The timer in Johnson is required in order to avoid the system hanging up due to a non-responding node. If a node does not respond with an ACK or NAK message, the timer will time out, which signals to the multicast sender node 12 to re-send the message via point-to-point communications (instead of multicast). We see from the express teachings of Johnson that the timer in Johnson is an integral part of the communication system, detecting when a non-responding node does not respond within a time period defined by the timer. Johnson thus represents the prior art discussed in appellant's Background of the Invention as quoted above. Because Johnson uses a timer to determine when a message has not been received by a non-responding node, and because Johnson uses a timer to determine that a data message has not been received by a non-responding node, Johnson expressly teaches away from the express limitations in claim 1.

Claim 1 includes the following limitation:

> . . . a job residing in the memory and executed by the at least one processor, the job processing the protocol, wherein the job functions according to receiver logic that uses the at least one ACK round to determine without using any timer whether the at least one data message has been received.

In rejecting the job recited in claim 1, the Examiner states that the messages are sent based on a date of birth, citing certain parts of Johnson. This language in the rejection shows a mismatch between the Examiner's logic and the claim limitation. The job recited in claim 1 functions according to *receiver logic* that uses at least one ACK round to determine without using any timer whether the at least one data message has been received. To state that the messages are sent based on date of birth shows the Examiner is considering the sender logic, not the receiver logic as recited in the claim.

In the Response to Arguments section of the final office action, the examiner references col. 2 lines 44-60 of Johnson as allegedly teaching the receiver logic in claim 1. Note, however, that the receiver logic in claim 1 "uses the at least one ACK round to determine without using any timer whether the at least one data message has been received." As stated clearly above, Johnson uses a timer while awaiting acknowledge signals, and therefore teaches away from the express limitations in claim 1.

Furthermore, the job in claim 1 processes the protocol, which specifies at least one acknowledge (ACK) round that provides a time benchmark for determining whether or not a message has been received without using any timer. The examiner has not identified in Johnson anything that reads on the job or the protocol in claim 1. The express limitations in claim 1 recite a protocol residing in the memory, and a job processing the protocol. Because the examiner has not identified corresponding teachings in Johnson that read on the protocol and job in claim 1, the examiner has failed to establish a prima facie case of anticipation for claim 1 under 35 U.S.C. §102(e).

For the many reasons given above, claim 1 is allowable over Johnson. Appellant respectfully requests that the examiner's rejection of claim 1 under 35 U.S.C. §102(e) be reversed.

Claims 5, 6, 10, 11, 15-18 and 22

Claims 5, 6, 10, 11, 15-18 and 22 are grouped with claim 1, and stand or fall according to the allowability of claim 1.

Claim 2

The arguments above with respect to claim 1 apply equally to claim 2, and are incorporated in this section by reference. Claim 2 recites:

> 2. The apparatus of claim 1 wherein the protocol comprises a plurality of phases that are each followed by an acknowledge (ACK) round.

In rejecting claim 2, the examiner cites to portions of Johnson that discuss data messages and acknowledge (ACK) messages. Nowhere does Johnson teach or suggest a protocol as recited in claim 1 that comprises a plurality of phases that are each followed by an ACK round, as recited in claim 2. In fact, the examiner's rejection does not indicate what teaching in Johnson allegedly reads on the protocol or the plurality of phases in claim 2. For this reason, the examiner has failed to establish a prima facie case of anticipation for claim 2 under 35 U.S.C. §102(e).

FIG. 8 of appellant's disclosure is an example of a protocol within the scope of claims 1 and 2. The fact that each phase is followed by an acknowledge round allows not having a timer. Johnson and other prior art use timers while awaiting ACK messages, as discussed in the Background of the Invention quoted above. Because Johnson uses a

timer while awaiting ACK messages, a protocol under Johnson would not need to have a plurality of phases that are each followed by an ACK round, as recited in claim 2. Johnson does not disclose a protocol as recited in claim 1, and does not disclose a protocol that comprises a plurality of phases that are each followed by an ACK round, as recited in claim 2. For these reasons, claim 2 is allowable over Johnson. Appellant respectfully requests that the examiner's rejection of claim 2 under 35 U.S.C. §102(e) be reversed.

Claims 7, 12, and 19

Claims 7, 12 and 19 are grouped with claim 2, and stand or fall according to the allowability of claim 2.

Claim 3

The arguments above with respect to claims 1 and 2 apply equally to claim 3, and are incorporated in this section by reference. Claim 3 recites:

> 3. The apparatus of claim 2 wherein each phase of the protocol is defined so that no node can both send a data message and receive a data message during any phase of the protocol.

In rejecting claim 3, the examiner states that Johnson recites the limitations in claim 3, citing col. 2 lines 18-34 of Johnson. Nowhere does the cited language in Johnson, or any other portion of Johnson, teach the limitations in claim 3. The examiner has not specifically identified what teaching in the cited portion of Johnson allegedly reads on the limitations in claim 3. As a result, the examiner has failed to establish a prima facie case of anticipation for claim 3 under 35 U.S.C. §102(e).

Johnson does not teach the protocol recited in claim 1. Johnson does not teach that the protocol of claim 1 comprises a plurality of phases that are each followed by an ACK round, as recited in claim 2. And Johnson does not teach that each phase of the protocol is defined so that no node can both send a data message and receive a data message during any phase of the protocol, as recited in claim 3. For these reasons, claim 3 is allowable over Johnson. Appellant respectfully requests that the examiner's rejection of claim 3 under 35 U.S.C. §102(e) be reversed.

Claims 8, 13, and 20

Claims 8, 13 and 20 are grouped with claim 3, and stand or fall according to the allowability of claim 3.

Claim 4

The arguments above with respect to claim 1 apply equally to claim 4, and are incorporated in this section by reference. Claim 4 recites:

> 4. The apparatus of claim 1 wherein the receiver logic comprises post-ACK logic that determines whether the at least one data message has been received after the job processes a selected one of the at least one ACK rounds.

In rejecting claim 4, the examiner states that Johnson teaches all of these limitations, citing col. 6, lines 33-46 of Johnson. Nowhere does the cited language in Johnson, or any other portion of Johnson, teach the limitations in claim 4. The examiner has not specifically identified what teachings in the cited portion of Johnson allegedly read on the limitations in claim 4. As a result, the examiner has failed to establish a prima facie case of anticipation for claim 4 under 35 U.S.C. §102(e).

Johnson does not teach the job with the receiver logic recited in claim 1. And Johnson does not teach that the receiver logic comprises post-ACK logic that determines whether the at least one data message has been received after the job processes a selected one of the at least one ACK rounds. In Johnson, a receiving node determines that a data message has been received, then sends an ACK message to the sender. The order in Johnson is thus backwards when compared to the order recited in claim 4. The order in Johnson expressly teaches away from the post-ACK logic in claim 4 that determines whether the at least one data message has been receiver ***after*** the job processes a selected one of the at least one ACK rounds. For these reasons, claim 4 is allowable over Johnson. Appellant respectfully requests that the examiner's rejection of claim 4 under 35 U.S.C. §102(e) be reversed.

Claims 9, 14, and 21

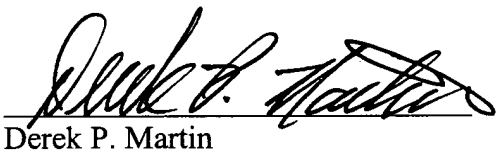
Claims 9, 14 and 21 are grouped with claim 4, and stand or fall according to the allowability of claim 4.

CONCLUSION

Claims 1-22 are addressed in this Appeal. For the numerous reasons articulated above, appellant maintains that the rejection of claims 1-22 under 35 U.S.C. § 102(e) is erroneous.

Appellant respectfully submits that this Appeal Brief fully responds to, and successfully contravenes, every ground of rejection and respectfully requests that the final rejection be reversed and that all claims in the subject patent application be found allowable.

Respectfully submitted,



Derek P. Martin
Reg. No. 36,595

MARTIN & ASSOCIATES, L.L.C.
P.O. Box 548
Carthage, MO 64836-0548
(417) 358-4700
Fax (417) 358-5757

APPENDIX - CLAIMS FINALLY REJECTED

1. An apparatus comprising:

at least one processor;

a memory coupled to the at least one processor;

a cluster engine residing in the memory and executed by the at least one processor, the cluster engine providing a mechanism for communicating ordered messages to and from a plurality of nodes in a computer cluster, wherein the apparatus comprises one node in the computer cluster;

a protocol residing in the memory that specifies at least one data message and at least one acknowledge (ACK) round that provides a time benchmark for determining whether or not a message has been received without using any timer; and

a job residing in the memory and executed by the at least one processor, the job processing the protocol, wherein the job functions according to receiver logic that uses the at least one ACK round to determine without using any timer whether the at least one data message has been received.

2. The apparatus of claim 1 wherein the protocol comprises a plurality of phases that are each followed by an acknowledge (ACK) round.

3. The apparatus of claim 2 wherein each phase of the protocol is defined so that no node can both send a data message and receive a data message during any phase of the protocol.

4. The apparatus of claim 1 wherein the receiver logic comprises post-ACK logic that determines whether the at least one data message has been received after the job processes a selected one of the at least one ACK rounds.

5. The apparatus of claim 1 wherein the receiver logic comprises pre-ACK logic that determines whether the at least one data message has been received before the job processes a selected one of the at least one ACK rounds.

6. A networked computer system comprising:

a cluster of computer systems that each includes:

a network interface that couples each computer system via a network to other computer systems in the cluster;

a memory;

a cluster engine residing in the memory that provides a mechanism for communicating ordered messages to and from the computer systems in the cluster;

a protocol residing in the memory that specifies at least one data message and at least one acknowledge (ACK) round that provides a time benchmark for determining whether or not a message has been received; and

a job residing in the memory and processing the protocol, wherein the job functions according to receiver logic that uses the at least one ACK round to determine without using any timer whether the at least one data message has been received.

7. The networked computer system of claim 6 wherein the protocol comprises a plurality of phases that are each followed by an acknowledge (ACK) round.

8. The networked computer system of claim 7 wherein each phase of the protocol is defined so that no node can both send a data message and receive a data message during any phase of the protocol.

9. The networked computer system of claim 6 wherein the receiver logic comprises post-ACK logic that determines whether the at least one data message has been received after the job processes a selected one of the at least one ACK rounds.

10. The networked computer system of claim 6 wherein the receiver logic comprises pre-ACK logic that determines whether the at least one data message has been received before the job processes a selected one of the at least one ACK rounds.

11. A computer-implemented method for processing a protocol using a plurality of jobs that form a group in a clustered computing environment, the method comprising the steps of:

providing a cluster engine for each member of the group that communicates with the other cluster engines in the group;

defining the protocol so that non-receipt of a message by the cluster engine can be determined without using any timer, the protocol including at least one data message and at least one acknowledge (ACK) round that provides a time benchmark for determining whether or not a message has been received; and

a job processing the protocol and functioning according to receiver logic that uses the at least one ACK round to determine without using any timer whether the at least one data message has been received.

12. The method of claim 11 wherein the step of defining the protocol further comprises the step of defining a plurality of phases that are each followed by an ACK round.

13. The method of claim 12 wherein the step of defining the protocol further comprises the step of defining each phase of the protocol so that no node can both send a data message and receive a data message during any phase of the protocol.

14. The method of claim 11 further comprising the step of the job using post-ACK logic that determines whether the at least one data message has been received after the job processes a selected one of the at least one ACK rounds.

15. The method of claim 11 further comprising the step of the job using pre-ACK logic that determines whether the at least one data message has been received before the job processes a selected one of the at least one ACK rounds.

16. A program product comprising:

(A) a protocol that specifies at least one data message and at least one acknowledge (ACK) round that provides a time benchmark for determining whether or not a message has been received without using any timer;

(B) a job that processes the protocol, wherein the job functions according to receiver logic that uses the at least one ACK round to determine without using any timer whether the at least one data message has been received; and

(C) computer-readable signal bearing media bearing the protocol and the job.

17. The program product of claim 16 wherein the signal bearing media comprises recordable media.

18. The program product of claim 16 wherein the signal bearing media comprises transmission media.

19. The program product of claim 16 wherein the protocol comprises a plurality of phases that are each followed by an acknowledge (ACK) round.

20. The program product of claim 19 wherein each phase of the protocol is defined so that no node can both send a data message and receive a data message during any phase of the protocol.

21. The program product of claim 16 wherein the receiver logic comprises post-ACK logic that determines whether the at least one data message has been received after the job processes a selected one of the at least one ACK rounds.

22. The program product of claim 16 wherein the receiver logic comprises pre-ACK logic that determines whether the at least one data message has been received before the job processes a selected one of the at least one ACK rounds.